

The Modular Aero-Propulsion System Simulation (MAPSS) Users' Guide

Khary I. Parker and Kevin J. Melcher
Glenn Research Center, Cleveland, Ohio

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at **<http://www.sti.nasa.gov>**
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA Access Help Desk at 301-621-0134
- Telephone the NASA Access Help Desk at 301-621-0390
- Write to:
NASA Access Help Desk
NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076



The Modular Aero-Propulsion System Simulation (MAPSS) Users' Guide

Khary I. Parker and Kevin J. Melcher
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center

Acknowledgments

The authors would like to thank Ten-Heui Guo, Jonathan Litt, Dan Simon, James Turso, Jacquelin Costrini, and Erin Aylward for the input they provided that went into the development of the software and this document.

This report is a formal draft or working paper, intended to solicit comments and ideas from a technical peer group.

Trade names or manufacturers' names are used in this report for identification only. This usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at <http://gltrs.grc.nasa.gov>

Table of Contents

Introduction	1
Nomenclature	1
Description of Model	2
Description of Key GUI Features.....	3
Overview.....	4
Editing Inputs.....	5
Assessing Model Output.....	7
Selecting Output Variables	8
Creating Linear Models	8
Custom Initial Condition Files.....	11
Loading and Saving Data.....	12
On-Line Help.....	12
Operation and Examples	12
Running a Basic Transient.....	12
Creating Linear Models	17
Concluding Remarks	18
References	18
Appendix A—Model Constants	19
Appendix B—Model Output Variables.....	21

The Modular Aero-Propulsion System Simulation (MAPSS) Users' Guide

Khary I. Parker and Kevin J. Melcher
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Introduction

The Modular Aero-Propulsion System Simulation (MAPSS) is a flexible turbofan engine simulation environment that provides easy access to health, control, and engine parameters through a graphical user interface (GUI). The key function of MAPSS is to provide the user with a graphical simulation environment in which to develop advanced control algorithms and quickly test them on a generic turbofan engine simulation. MAPSS can be used to generate state-space linear models, from which the user may create a piecewise linear controller. MAPSS can also run transient simulations. This capability allows the user to test the performance of their controller on a validated, and verified, generic engine model. This is a simple example of what MAPSS can do.

The engine model used in this environment is based on a low frequency, transient, performance model of a high-pressure ratio, dual-spool, low-bypass, military-type, variable cycle, turbofan engine with a digital controller. The GUI is used to input model parameters, such as power lever angle (PLA), Mach number, and altitude. In addition, the GUI will allow changes to controller and engine constants, as well as other parameters that may be varied during control development. Desired output variables are user selectable to aid in analysis of engine and control parameters. The output data as well as any changes to parameters and constants may be saved and reloaded into the GUI later. A more detailed discussion on the development of MAPSS may be found in [1].

This guide begins by describing the engine model provided with MAPSS. This is followed by a description of the graphical user interface (GUI) that is provided to facilitate user interaction with the simulation. The GUI allows the user to specify input, select and access outputs, create linear models, and to save both input and output data. The guide concludes by presenting two step-by-step examples for running MAPSS. Note that this guide assumes that the reader has a basic understanding of the commands, scripting, and terms associated with the MATLAB [2]/Simulink [3] software.

Nomenclature

A8	nozzle exit area
A16	bypass duct exit area
AB	afterburner
alt	altitude
AltvTM	altitude versus time profile
CAD	controller and actuator dynamics
CLM	component level model
DeciVal	output decimation value variable
Decrip	run description variable
GUI	graphical user interface
HPC	high-pressure compressor
HPT	high-pressure turbine

LPT	low-pressure turbine
MAPSS	modular aero-propulsion system simulation
PLA, pla	power lever angle
PLAvTM	PLA versus time profile
SimDur	simulation duration variable
TMPC	average hot section temperature
VABI	variable area bypass injector
WF36	burner fuel flow
xm	Mach number
XNL	low-pressure rotor spool speed
XNH	high-pressure rotor spool speed

Description of Model

MAPSS is a non-real time, multi-rate simulation of a modern high-pressure ratio, dual-spool, low bypass, variable cycle, military-type engine, with a digital controller. The model is composed of the “Controller and Actuator Dynamics” (CAD) and “Component Level Model” (CLM) modules. A basic block diagram of the simulation is shown in figure 1. The controller model in the CAD module emulates the functionality of a digital controller, which has a typical update rate of 50 Hz. The CLM module simulates the dynamics of the engine components and uses an update rate of 2500 Hz. The higher update rate is required to allow equations that compute mass-energy balances to converge. The actuator models in the CAD module have the same update rate as the CLM.

The components simulated in the MAPSS CLM (fig. 2) are the fan, booster, high-pressure compressor (HPC), burner, high- and low-pressure turbines (HPT and LPT), mixer, afterburner (AB), and nozzle. The fan inlet temperature and pressure are external inputs based on user-defined operating conditions. The CLM also models the forward blocker door area positioning (A14), the aft variable area bypass injector (VABI) area positioning (A16), the flow through the bypass duct, and the bleed flows. The bypass duct module determines the pressure, temperature, enthalpy, and flow rate up to the mixing plane. The CLM does not include an inlet model because it is typically not considered part of the engine. In addition to these parameters, the compressor bleeds and cooling flows are simulated.

The three state variables used in this model are low-pressure rotor speed (XNL), high-pressure rotor speed (XNH), and the average hot section metal temperature (TMPC) (measured from aft of the combustor to HPT). Inputs into the CLM from the controller model consist of ambient conditions and fan inlet temperature and pressures, and actuator commands. The CLM outputs are the calculated values and simulated sensed output values from the sub-modules that comprise individual component parameters (i.e. temperature, pressure, gas flow, etc.).

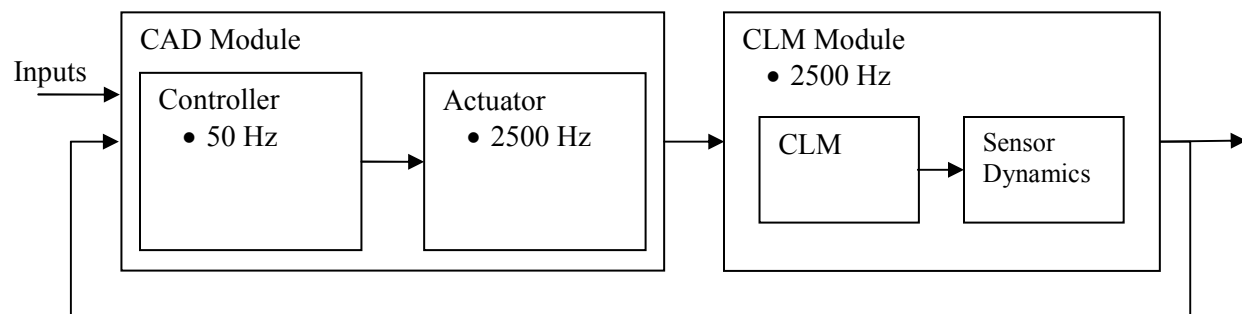


Figure 1: Basic block diagram of MAPSS.

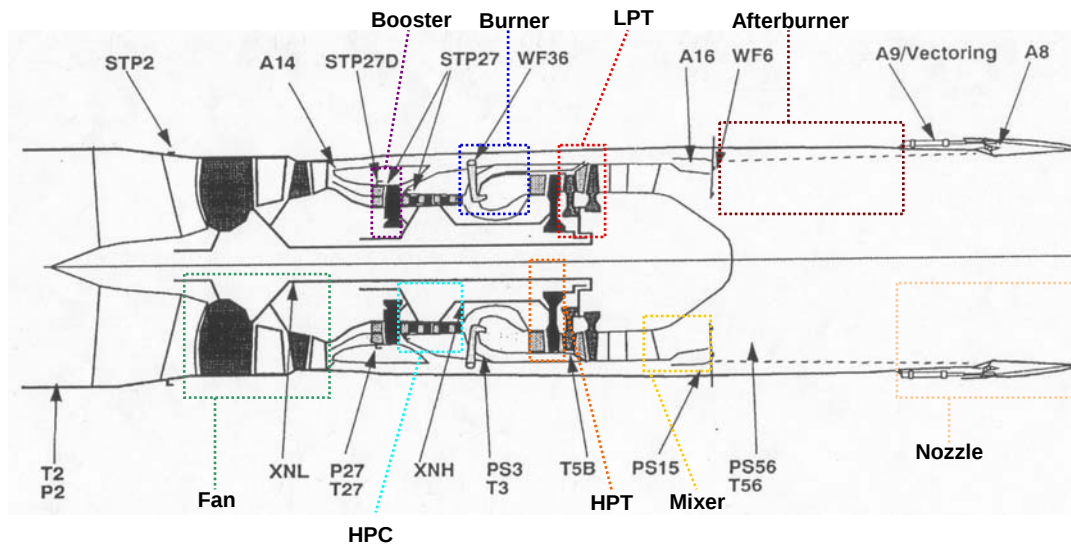


Figure 2: Schematic of MAPSS CLM.

The CAD module contains the controller and the actuator dynamics sub-modules. The controller sub-module uses calculated and simulated sensed output values from the CLM to determine if the desired set points have been reached. The set points are based on the user defined operating conditions of PLA, altitude, and Mach number. The controller uses maps to schedule the actuator demands for the fan, booster, and HPC stator positions using the engine's operating conditions. To obtain the actuator demands for the three control parameters, burner fuel flow (WF36), nozzle exit area (A8), and bypass exit area (A16), the controller uses five different modes, with a regulator for each mode: high power, medium power, low power, fan stall margin, and overspeed. The high power mode handles CLM operation for PLA values larger than 42.5° . The medium power mode handles CLM operation for PLA values between 32.5° and 37.5° . The low power mode handles CLM operation for PLA values below 32.5° . As the controller transitions between the three power modes, percentages of each regulator output, based on the current value of PLA, fan stall margin, and XNL, are added together to obtain the final actuator demand. This is a technique known as blending. In addition to the power modes there is the fan stall margin mode which controls the fan in the stall region, and the overspeed mode which limits XNL. These last two modes may be turned on and off upon request.

The actuator dynamics sub-module simulates the dynamics of the torque motors, servomechanisms and feedback to determine actuator positions for the fan, HPC, and booster guide vanes; nozzle, mixer, and VABI areas; and fuel flow actuator. Lags are used to simulate actuator dynamics, which are also captured in MAPSS.

Description of Key GUI Features

A large number of model parameters must be defined in the MATLAB workspace prior to running a MAPSS simulation. To facilitate setup of these parameters and operation of MAPSS simulations, the graphical user interface (GUI) shown in figure 3 was implemented. This section describes the features of the GUI that can be used to setup and run MAPSS simulations.

Overview

The MAPSS GUI manages the input/output information for MAPSS. It has three main sections: MODEL INPUTS, RESULTS LIST, and MODEL OUTPUTS.

The MODEL INPUTS section displays the operating conditions given by the setup script file *mapss.m*, as well as providing a means to change model constants and simulation run parameters. The RESULTS LIST displays a record of operating conditions for each completed MAPSS run. The MODEL OUTPUTS section allows the user to select model output variables for each run. Output generated by the model along with user defined input information is stored in a structured output array that may be saved as a MATLAB MAT-file, which can be reloaded into the GUI or accessed independently. In addition, the MAPSS GUI can be used to create custom non-zero initial condition start files, generate linear models from the CLM at specified steady-state operating points, and plot the time response of selected variables for each of the runs shown in the Results List.

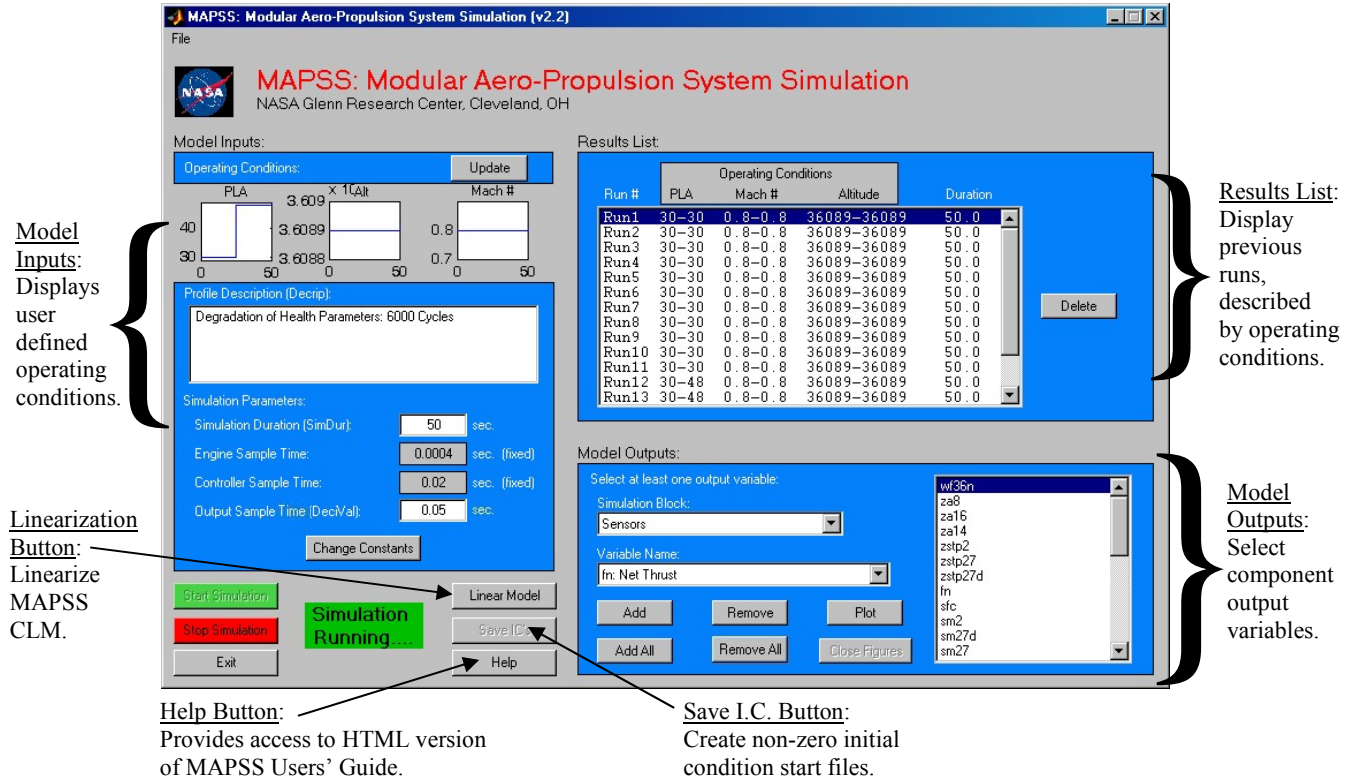


Figure 3: MAPSS graphical user interface.

Editing Inputs

A MATLAB M-file startup script, called *mapss.m*, is used to load some input data into the MAPSS GUI. This script allows the user to modify the simulation's operating condition profiles: power lever angle (PLA), altitude, and Mach number; as well as provide a description of the simulation run, steady-state operating points for linearization, simulation duration, and output sample time. Figure 4 shows an example of the *mapss.m* script.

The MODEL INPUTS section of the MAPSS GUI (fig. 5) displays the operating conditions, the current run description, and simulation parameters that were entered in *mapss.m*. All fields and axes in this section obtain their data from the MATLAB workspace when the *mapss.m* script is initially executed. Subsequent updates occur when the edited script is re-executed and the UPDATE button is clicked.

```

1  % mapss.m
2  % This start-up script file provides a means to enter user-defined input information into
3  % the MATLAB workspace to initialize the Modular Aero-Propulsion System Simulation (MAPSS).
4  %
5
6  % Operating Condition Profiles
7  pla = 21;
8  PLAvTM = [0.00 1.00 1.02 5.00 10.00 99.00;
9            0+pla 0+pla 5+pla 5+pla 5+pla 5+pla]; % Time vs. Power Lever Angle (degrees)
10
11 alt = 0;
12 AltvTM = [0.00 1.00 3.00 5.00 10.00 15.00 99.00;
13           0+alt 0+alt 5000+alt 5000+alt 5000+alt 5000+alt 5000+alt]; % Time vs. Altitude (feet)
14
15 xm = 0.0;
16 XMvTM = [0.00 1.00 2.00 5.00 10.00 15.00 99.00;
17          0+xm 0+xm 0.2+xm 0.2+xm 0.2+xm 0.2+xm 0.2+xm]; % Time vs. Mach #
18
19 % Provide Description of Run (DO NOT CHANGE THE NAME OF THIS VARIABLE)
20 Decrip = 'Ground Idle Conditions';
21
22 % Simulation Parameters
23 SimDur = 10; % Simulation Duration (sec)
24 DeciVal = 0.05; % Output Sample Rate
25
26 % Linearization Operating Points
27 linmod_pla = [25 30];
28 linmod_alt = [0 6000];
29 linmod_xm = [0 0.3];
30 linmod_QUIVal = [0.1 0.1 0.001]; % Quiescence Value:
31 % [LP Spool Speed, HP Spool Speed, Avg. Metal temperature]
32
33 %----- DO NOT EDIT -----
34 if ~(exist('onoff'))
35     mapss_gui;
36     edit mapss;
37 end
38 %----- DO NOT EDIT -----

```

Figure 4: Example of a *mapss.m* script file.

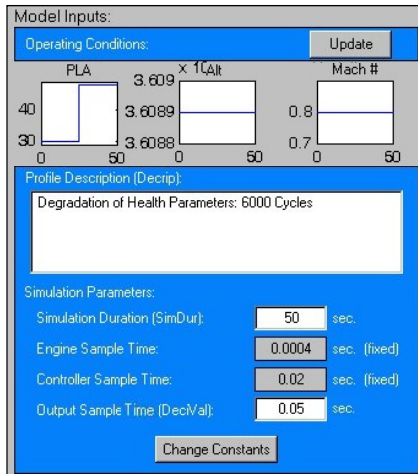


Figure 5: Example of a model inputs section.

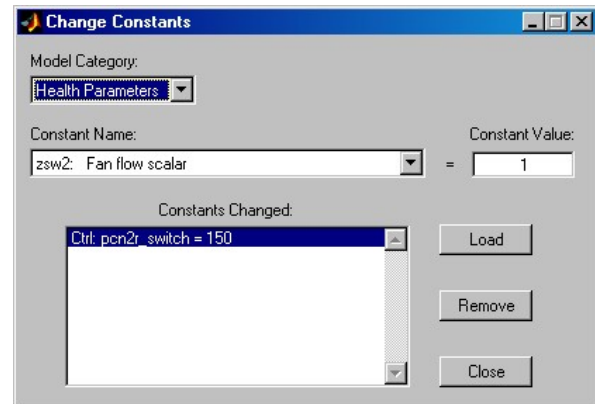


Figure 6: Example of a change constants GUI.

The OPERATING CONDITIONS sub-section displays user defined time histories for PLA, altitude, and Mach number obtained from the script file. It provides a means of visually verifying the user specified input operating conditions. The PROFILE DESCRIPTION (Decrip) sub-section displays the current run description. The SIMULATION PARAMETERS sub-section displays the SIMULATION DURATION, ENGINE SAMPLE TIME, CONTROLLER SAMPLE TIME, and OUTPUT SAMPLE TIME. The SIMULATION DURATION (SimDur) is the time, in simulation seconds, at which MAPSS will terminate a given run. The CONTROLLER SAMPLE TIME and the ENGINE SAMPLE TIME are fixed values of 0.02 seconds and 40 milliseconds, respectively. Any change to these values may cause the simulation to generate incorrect results. The OUTPUT SAMPLE TIME (DeciVal) specifies the sample at which MAPSS will save the generated output data. The decimation is based on the ENGINE SAMPLE TIME. Data in the PROFILE DESCRIPTION and editable parts of the Simulation Parameters may be entered directly into the MAPSS GUI or from the parameters defined in the *mapss.m* script. The MAPSS GUI looks for variables Decrip, SimDur and DeciVal in the MATLAB workspace when assigning strings and values for PROFILE DESCRIPTION, SIMULATION DURATION, and OUTPUT SAMPLE TIME, respectively.

There are many constants used in this model that the user may desire to change. Clicking the CHANGE CONSTANTS button activates a sub-GUI (fig. 6) that allows the user to change these constants.

Instruction for changing the model constants via this GUI are given in the Operations and Examples section and a list of all available constants are given in Appendix A. Model constants are grouped into five categories. Selecting a specific category from the drop-down list provides the user with access to constants in that category.

The constants may also be changed at the command line via the structure array Const. This array contains every constant used by MAPSS; so, caution should be used when editing the data in this structure array.

Once all the input parameters are set, MAPSS may be run by clicking on the SIMULATION START button.

Accessing Model Output

After each run, the MAPSS GUI will display operating condition information in the Results List and create a structure array in the MATLAB workspace called **MAPSSoutput**. Each run is stored as a new element in **MAPSSoutput**. Each element of this structure contains fields that store the user defined operating conditions, run description information, simulation parameters, model output data, model constants, and component maps. The purpose of this structure is to give the user access to the model generated data from both the MAPSS GUI and the MATLAB command line. Figure 7 shows an example of the data structure within an element of **MAPSSoutput**.

All available output data is stored in the **mdlOut** field. The number of columns corresponds to the number of variables, and the number of rows corresponds to the number of data points saved.

All the runs stored in the existing **MAPSSoutput** variable are shown in the RESULTS LIST subsection from the MAPSS GUI (fig. 8). The RESULTS LIST displays some basic information about each run, such as run number, operating condition (PLA, Mach number, and altitude) and the duration.

Next to the list box is the DELETE button, which is used to remove a run from the RESULTS LIST and from the **MAPSSoutput** variable. Only one run can be removed at a time.

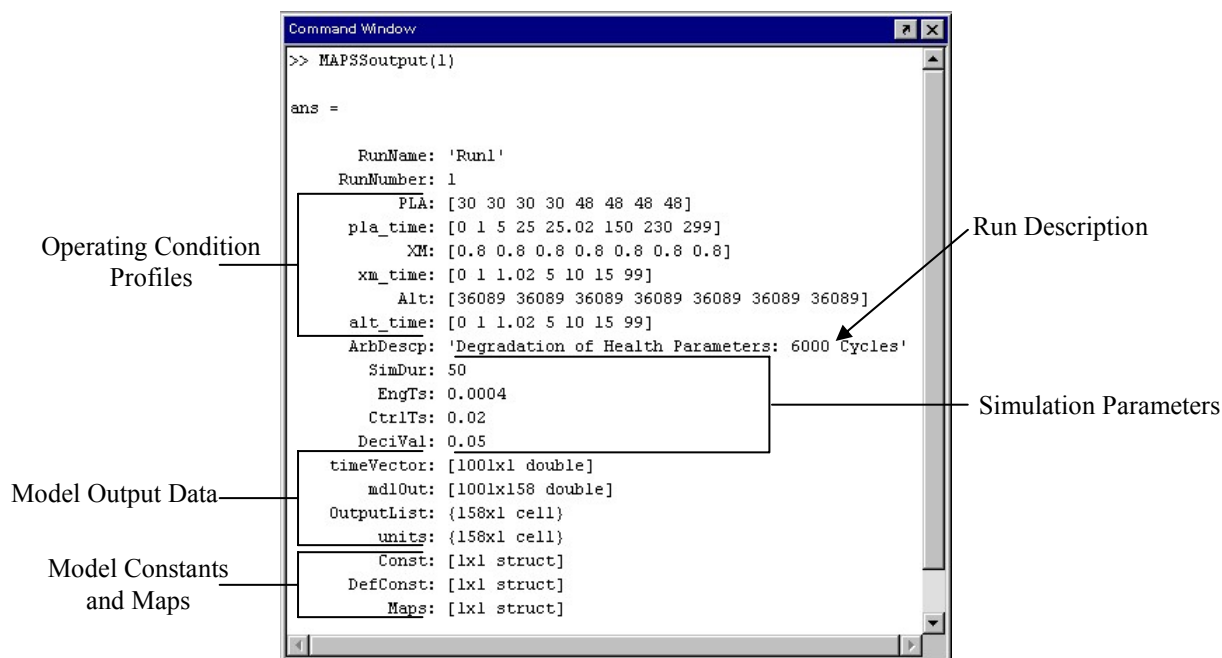


Figure 7: Example of MAPSSoutput data fields.

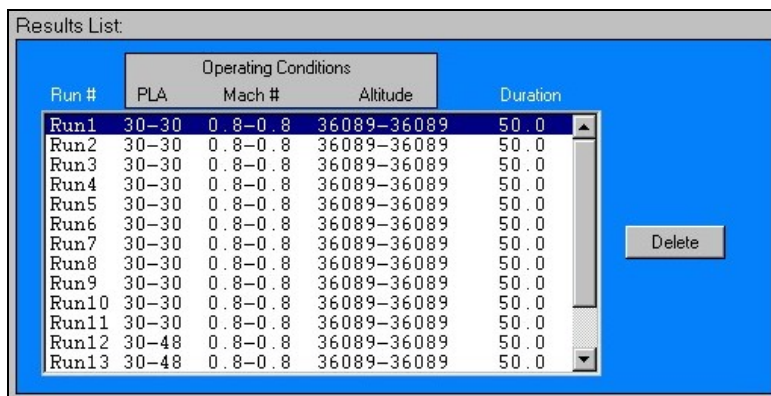


Figure 8: Example of a results list.

Selecting Output Variables

The MODEL OUTPUTS section (fig. 9) allows the user to select any of the available model output variables. The list of these variables can be found in Appendix B. Instruction for selecting output variables will be given in Operations and Examples section.

The output variables may be chosen by selecting a simulation block name from the SIMULATION BLOCK drop-down list. The block names in this list correspond to the block names in the model. This list has three categories: CAD OUTPUT, COMPONENT OUTPUT, and SENSOR & STATE OUTPUT. Selecting one of these categories from the drop-down list provides the user with access to the output variables for the selected simulation block. The CAD OUTPUT category contains output variables for the actuator positions and input temperatures and pressures (e.g. ambient and fan inlet). The COMPONENT OUTPUT category contains output variables from each of the CLM components, values for power consumption and power output, and values for the update parameters; i.e., parameters used to update the CLM after each time step, ensuring a balanced CLM prior to the next control command. The SENSOR & STATE OUTPUT category contains the output variables for each state and sensor in the model.

Selected variables are added to the list box on the right. If there is at least one run in the RESULTS LIST and at least one output variable selected, the user may click the PLOT button to plot data. The PLOT button is used to generate time history plots of the user selected output variable data for the highlighted run. Each plot figure window will display up to four (4) subplots of output data before another figure window is created. The CLOSE FIGURES button will close all the plot figure windows that were opened by the PLOT button.

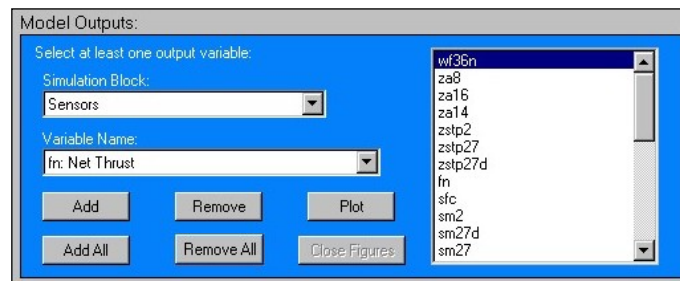


Figure 9: Example of a model outputs section.

Creating Linear Models

The MAPSS GUI allows the user to create a linear model of the CLM about a user-defined steady-state operating point in the form of state space matrices A, B, C, and D. The linearization routine also produces the associated input, output, and state trim values as well as storing the nonlinear data associated with the given operating point. The nonlinear data is stored in the event the linear model needs to be validated. The state-space matrices and trim values are used to create the following linear input-output relationship:

$$\begin{aligned}\Delta\dot{x} &= A\Delta x + B\Delta u \\ \Delta y &= C\Delta x + D\Delta u\end{aligned}$$

where $\Delta\dot{x} = \dot{x} - \dot{x}_{\text{trim}}$, $\Delta u = u - u_{\text{trim}}$, $\Delta y = y - y_{\text{trim}}$. Figure 10 is a graphical representation of the above equations.

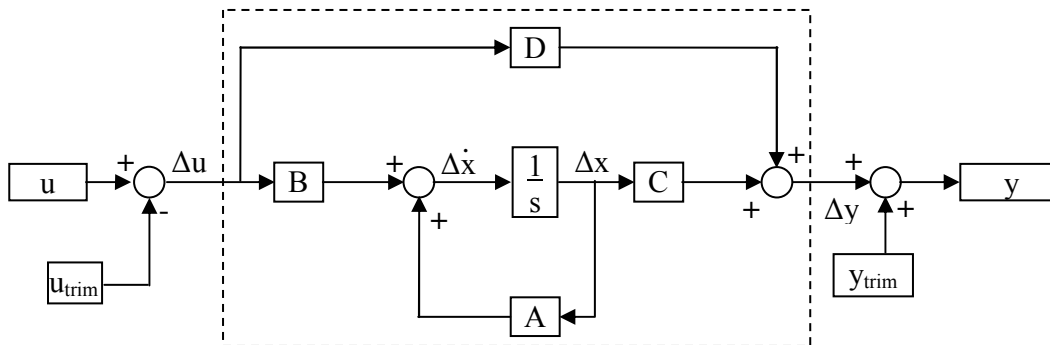


Figure 10: State-space representation of linear engine model.

```

26 % Linearization Operating Points
27 - linmod_pla = [25];
28 - linmod_alt = [0 6000];
29 - linmod_xm = [0 0.3];
30 - linmod_QUIVal = [0.1 0.1 0.001];

```

Figure 11: Example of a linearization script from *mapss.m*.

The output trim and nonlinear output values, Y , are the calculated and sensed output values from the CLM (i.e., “CLM Sensor”). The state trim values, X , are the low-pressure rotor speed, high-pressure rotor speed, and the average hot section metal temperature. The input trim values, U , are the parameters determined by the PI control action in the controller: burner fuel flow, nozzle exit area, and bypass exit area.

The operating points are entered in the Linearization Operating Points section of *mapss.m* (fig. 11).

The user may enter as many steady state operating points as desired. The only caveat is that the initial condition data file for the first point must exist in a directory. Creating these files will be explained in the next section. The vector `linmod_QUIVal` contains the values used by a quiescence function to determine when the simulation terminates. These values correspond to the smallest value that the state derivatives (XNL, XNH, and TMPC respectively) must reach before linearization of the CLM begins.

When the user clicks on the LINEAR MODEL button in the lower left portion of the GUI



Figure 12: Example of a LINEAR MODEL button.

two Simulink MDL-files open: *mapss_mdl.mdl* (nonlinear model) and *mapss_linmod.mdl* (linearization model). In case the user made any changes to the CLM, MAPSS takes a copy of the CLM from *mapss_mdl.mdl* and places it into *mapss_linmod.mdl* to ensure that the linear models contain the most current information of the CLM. MAPSS will run the nonlinear model to steady state, where a quiescence function determines when the state derivatives are sufficiently small enough to terminate the nonlinear model, based on all three limits set in `linmod_QUIVal` being true. At the end of the run, all the data at the final time step is stored for later use and to assist in linearizing the CLM. MAPSS uses the MATLAB commands *trim* and *linmod* to create a linear model of the CLM in *mapss_linmod.mdl*. In the MATLAB command window, MAPSS will display the operating condition for each linear model that has been created (fig. 13).

```

>> mapss

Linearize operating point: PLA = 21, Altitude =    0 ft., Mach # = 0.00
State-space matrices created

Linearize operating point: PLA = 21, Altitude =    0 ft., Mach # = 0.30
State-space matrices created

Linearize operating point: PLA = 21, Altitude = 6000 ft., Mach # = 0.00
State-space matrices created

Linearize operating point: PLA = 21, Altitude = 6000 ft., Mach # = 0.30
State-space matrices created

Linearization complete

```

Figure 13: Example of a MAPSS linearization display.

Name	Size	Bytes	Class
MAPSS_A	3x3x4	288	double array
MAPSS_B	3x3x4	288	double array
MAPSS_C	22x3x4	2112	double array
MAPSS_D	22x3x4	2112	double array
MAPSS_QUIVal	1x3	24	double array
MAPSS_Utrim	3x1x4	96	double array
MAPSS_Xtrim	3x1x4	96	double array
MAPSS_Ysensor	2407x23x4	1771552	double array
MAPSS_Ytrim	22x1x4	704	double array
Grand total is 222159 elements using 1777272 bytes			

Figure 14: Example of linearization output variables.

If multiple operating conditions are entered, MAPSS will run a step transient, from the previous operating point to the new operating point, to steady state. The step will occur at 1.02 seconds. MAPSS cycles through the operating points using a nested loop routine, where the Mach number is in the inner loop, altitude is in the middle loop, and PLA is in the outer loop. The advantage to this method is to give the user a straightforward means to generate piecewise linear models within the engine's flight envelope.

At the end of the run, the A, B, C, D matrices, as well as input, output, state trim variables and the time history output are created and stored in the MATLAB workspace (fig. 14).

The general storage format for each matrix is a three-dimensional array, except for the vector MAPSS_QUIVal. The first two dimensions are the size associated with the A, B, C, and D state-space matrices. The third dimension represents the number of operating conditions for which a linear model has been generated. Columns of the MAPSS_Yout variable contains time history data for each of the twenty-two CLM output sensor variables, plus one column for time.

Custom Initial Condition Files

MAPSS is supplied with two non-zero start files containing initial conditions for the model that represent the following operating conditions:

- 1) Ground Idle: PLA = 21, Altitude = 0, Mach = 0
- 2) Cruise: PLA = 35, Altitude = 35000, Mach = 0.8

Other operating points may be generated as follows:

- Define a step transient from an existing operating point to the desired operating point in *mapss.m*.
- Run the transient in MAPSS. The user should be especially careful about running the model long enough to ensure that all state parameters have reached steady-state.
- Click the SAVE IC'S button to generate a non-zero start file at the new operating point.



Figure 15: Example of a SAVE IC's button.

This will create a file that MAPSS will be able to identify when this operating point is used as a starting point. The file name will be formatted as follows:

Mach = 0.20
mapss_nzstrt_pla260_m020_a05000.mat
PLA = 26.0 Altitude = 5000

The file will be saved in the current working directory. When MAPSS is run, it will search for a file, with the above format, corresponding to the user-defined operating conditions. MAPSS will generate an error window if the file does not exist.

If the same operating point is run to steady state and the “Save IC’s” button is clicked again (assuming you are in the same directory), a window will pop up asking if you want to overwrite the current file.

Click “Overwrite”, the default selection, to overwrite the pre-existing file. Otherwise, click “Cancel”.

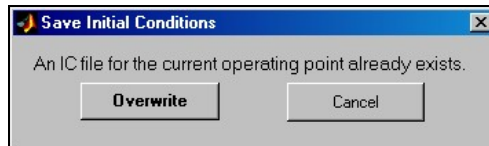


Figure 16: Example of a overwrite dialog box.

Loading and Saving Data

The data created by MAPSS may be saved and reloaded from the MAPSS FILE menu.

The SAVE and SAVE AS... selections will allow the user to save the MAPSSoutput structure as a MAT-file. The SAVE AS... option will open a SAVE dialog window where the MAT-file can be named. If the data was not previously saved, the SAVE option will be grayed-out. To help differentiate the saved data file from other MAT-files that may be in the current working directory, it is suggested that the user give the file name some type of prefix (i.e., MAPSS_*.mat).

To load data, click on the LOAD DATA... option. This will open an OPEN dialog window where the MAT-file can be found and loaded into the MATLAB workspace, and fill the RESULTS LIST with information about the data in the MAT-file.

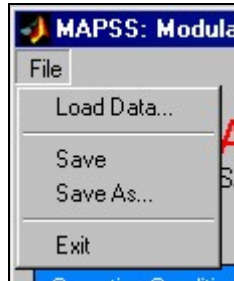


Figure 17: Example of a file menu.

On-Line Help

An HTML version of this Users' Guide is available via the MAPSS GUI. To access on-line help, click the "Help" button. Various topics can be reviewed via links in the Table of Contents to the various sections of this guide.

Operation and Examples

The MAPSS GUI was designed to provide the user with a straightforward means to conduct individual runs of the MAPSS model for various operating points throughout the flight envelope. This section will give a step-by-step explanation on how to run MAPSS via two examples: running a basic transient with changing health parameters, and creating a linear model. Note that the values used in these examples are for illustrative purposes only and are not meant to represent typical engine operations.

Running a Basic Transient

This example will demonstrate how to set up a transient run in MAPSS and make changes to the model's health parameters. Consider a run where the PLA is stepped from 21 to 30 at 1.02 seconds; while the altitude is increased from 0 to 5000 over two seconds; and the Mach number is increased from 0 to 0.2 over one second. The HPT has decreased in efficiency by 15%. The simulation will run for 20 seconds and store data every 0.5 seconds. The goal is to determine the behavior of the actuators.

The procedure for implementing and running the example simulations is as follows:

- 1) Type `mapss` at the MATLAB command prompt. This will open the `mapss.m` script and the MAPSS GUI.
- 2) In the `mapss.m` script, edit the time and data profiles for the variables `pla`, `xm`, `alt`, `PLAvTM`, `XMvTM`, and `AltvTM`. The variables `pla`, `xm`, and `alt` are the initial points; `PLAvTM`, `XMvTM`, and `AltvTM` are the profiles based on those initial points. Define desired time histories for each of these input variables. Figure 18 shows how variables would be defined for the given example.

```

6  % Operating Condition Profiles
7  - pla = 21;
8  - PLAvTM = [0.00 1.00 1.02 5.00 10.00 99.00;
9            0+pla 0+pla 5+pla 5+pla 5+pla 5+pla]; % Time vs. Power Lever Angle (degrees)
10
11 - alt = 0;
12 - AltvTM = [0.00 1.00 3.00 5.00 10.00 15.00 99.00;
13           0+alt 0+alt 5000+alt 5000+alt 5000+alt 5000+alt 5000+alt]; % Time vs. Altitude (feet)
14
15 - xm = 0.0;
16 - XMvTM = [0.00 1.00 2.00 5.00 10.00 15.00 99.00;
17           0+xm 0+xm 0.2+xm 0.2+xm 0.2+xm 0.2+xm 0.2+xm]; % Time vs. Mach #

```

Figure 18: Operating condition profiles from `mapss.m`.

- 3) Name this run “Basic Transient” by editing the PROFILE DESCRIPTION (Decrip) string.
- 4) Set SIMULATION PARAMETERS (SimDur) equal to 20 and OUTPUT SAMPLE TIME (DeciVal) equal to 0.5.
- 5) Save and run `mapss.m` to load the new variable definitions into the MATLAB workspace.
- 6) In the MAPSS GUI, click the “Update” button to refresh the operating condition displays.
- 7) To change the health parameters, first click on the CHANGE CONSTANTS button to open the Change Constants sub-GUI:
 - i. From the MODEL CATEGORY drop-down menu, select HEALTH PARAMETERS.

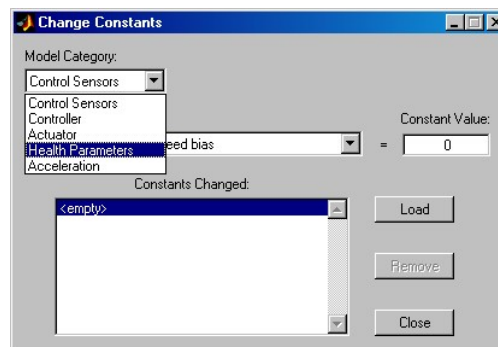


Figure 19: Select health parameters from list.

This will load the names for all available health parameters into the CONSTANT NAME drop-down menu. A name and description has been associated with the constants found in the MAPSS simulation.

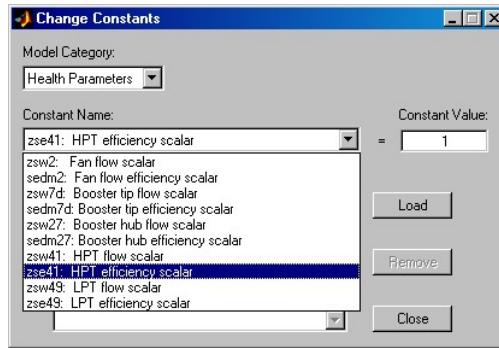


Figure 20: Select constant from list.

- ii. The current value of the each constant, when selected, will appear in the CONSTANT VALUE text box. Select the variable **zse41** from the Constant Name menu then change its value to 0.85.

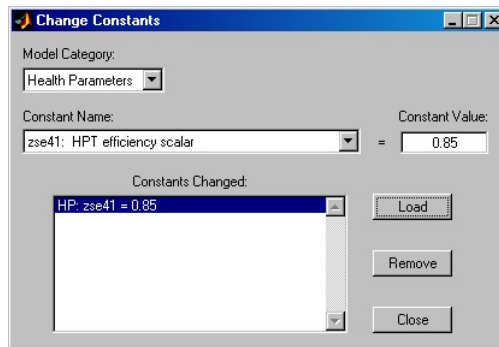


Figure 21: constants changed.

- iii. Click the “Load” button. This will load the new value for **zse41** in the workspace. In the CONSTANTS CHANGED window, the category, constant name, and its new value will appear. This will help the user keep track of which constants were changed. To restore **zse41** back to its default value, select it from the CONSTANTS CHANGED window and click REMOVE.
 - iv. When finished with the sub-GUI, click CLOSE.
- 8) To start the simulation click on the START SIMULATION button. This will open the *mapss_md1.mdl* nonlinear engine model, two scopes for monitoring the state variables and their derivatives, and a status bar. The status bar gives the remaining simulation time, percent complete, and the ratio of actual time to simulation time.

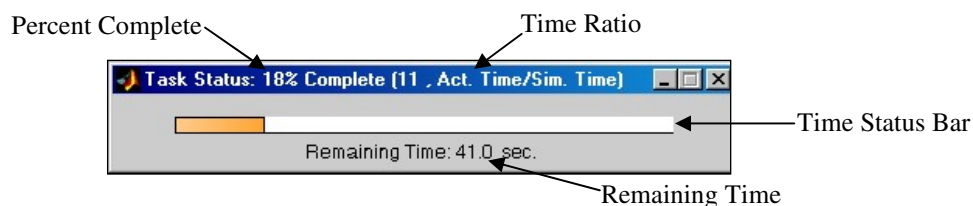


Figure 22: MAPSS run status bar.

Before *mapss_md1.mdl* runs, MAPSS will search the current directory for the non-zero start file

`mapss_nzstrt_pla210_m000_a00000.mat`

If it exists, the file will be loaded into the MATLAB workspace and the model will be run. If the file does not exist, an error GUI will appear stating that the file does not exist in the current directory and the model will not run.

- 9) MAPSS will exit the simulation mode automatically when the simulation time reaches the value of SimDur. However, the user may stop the simulation at anytime during the run by clicking the STOP SIMULATION button. At the completion of the run or if it is stopped, MAPSS will display the operating condition in the Results List and create the MAPSSoutput structure array in the MATLAB workspace.
- 10) Select output variables from the MODEL OUTPUTS section of the MAPSS GUI. For this example, select all the actuator position variables as output:
 - i. Click on the SIMULATION BLOCK drop-down menu and select ACTUATOR POSITIONS.

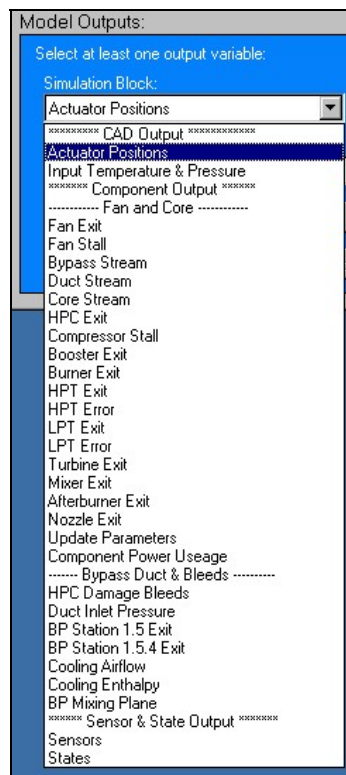


Figure 23: Select simulation block from listing.

This will load all available output variables into the VARIABLE NAME drop-down menu.

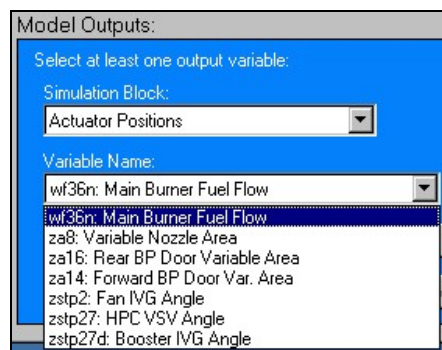


Figure 24: Select variable from listing.

- ii. From the VARIABLE NAME drop-down menu, select “WF36N: MAIN BURNER FUEL FLOW”, then click the ADD button. This will add the variable to the output list. Select another variable from the drop-down menu then click the ADD button again. Continue this process until all the variables in the drop-down menu appear in the output list.



Figure 25: Variables added to the output list.

To add the variables in the drop-down menu for a given SIMULATION BLOCK simultaneously, click the ADD ALL button. If a particular variable needs to be removed (e.g. “WF36N: MAIN BURNER FUEL FLOW”), click on the variable name in the output list and click REMOVE. To clear the entire list click CLEAR ALL.

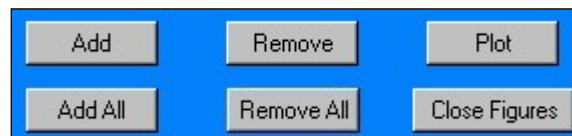


Figure 26: MODEL OUTPUTS buttons.

- 11) Select the desired run from the RESULTS LIST. Plot the time histories of all variables listed in the MODEL OUTPUTS by clicking on the PLOT button. The data can also be accessed from the command line. For example, to plot burner fuel flow, at the MATLAB command prompt type:

```
>> plot(MAPSSoutput.timeVector, MAPSSoutput.mdlOut(:,1));
```

See Appendix B for a listing of all available output parameters.

Close all figures containing time history plots generated by the MAPSS GUI by clicking on the CLOSE FIGURES button in the RESULTS LIST section.

- 12) Save the current model conditions in a non-zero start file by clicking the Save IC's button. The new initial conditions will be used in a subsequent example on model linearization. The simulation in this example requires about 100 seconds for the model to reach steady state conditions. However, for the purpose of educational instruction, this pseudo-steady point is assumed to be sufficiently accurate.
- 13) To close MAPSS, click on the “Exit” button, or go to the “File” menu and click on “Exit”.

Creating Linear Models

This example will demonstrate how to create a linear model. Suppose a linear model is needed at each of the following operating conditions, starting with the initial operating point given in the last example: PLA at 25 and 30, altitudes at 0 and 6000, and Mach numbers at 0 and 0.3.

- 1) Type `mapss` at the MATLAB command prompt. This will open the `mapss.m` script and the MAPSS GUI (if not already open).
- 2) In the `mapss.m` script, enter the desired PLA operating points into the vector `linmod_pla`, altitude operating points into the vector `linmod_alt`, Mach number operating points into the vector `linmod_xm`, and quiescence values into the vector `linmod_QUIVal` as shown in figure 27.

```
26 % Linearization Operating Points
27 - linmod_pla = [25 30];
28 - linmod_alt = [0 6000];
29 - linmod_xm = [0 0.3];
30 - linmod_QUIVal = [0.1 0.1 0.001];
```

Figure 27: Linearization script from `mapss.m`.

- 3) Setting a value for the Output Sample Time (`DeciVal`) is not necessary because no data will be entering MAPSSoutput.
- 4) Run `mapss.m` to load the variables into the MATLAB workspace.
- 5) In the MAPSS GUI, click the LINEAR MODEL button; this will begin the linearization process.
 - i. MAPSS will take the first operating point, PLA = 21 degrees, alt = 0 feet, xm = 0 (also written as [21, 0, 0]), and run the nonlinear model to steady state.
 - ii. At the end of the run, all of the model output data at the final time step is stored for later use and will be used in linearization. From the stored data set, MAPSS will extract the actuator positions, input temperatures and pressures, final sensor measurements, and states to create the linear model.
 - iii. With this data, MAPSS uses the MATLAB `trim` command to determine the trim parameters for linearization and the `linmod` command to create the A, B, C, D matrices.
 - iv. All the linearization, trim and nonlinear sensor data, with time, are then stored in local variables until all operating points have been linearized.
 - v. When the first linear model has been created, the next operating point will be [21, 0, 0.2]. MAPSS will run a step transient from the previous operating point to the new operating point using the data from the stored nonlinear data set as initial conditions. The step to the new point will occur at 1.02 seconds. The cycle will begin again with step 5.i.
 - vi. When the linear model for this operating point has been created, the next point will be [21, 3000, 0]. MAPSS will continue with steps 5.i to 5.vi until linear models of all combinations have been created.
- 6) When MAPSS has completed the linearization of all the operating points, the MATLAB workspace will contain the following linearization output variables: The trim results are stored in the arrays `MAPSS_Xtrim`, `MAPSS_Utrim`, and `MAPSS_Ytrim`; the linearization results are stored in `MAPSS_A`, `MAPSS_B`, `MAPSS_C`, and `MAPSS_D`; and the nonlinear sensor output results, with a time vector, are stored in `MAPSS_Ysensor`; and the quiescence values are stored in `MAPSS_QUIVal`.

Concluding Remarks

This document is the first version of a users' guide for the Modular Aero-Propulsion System Simulation (MAPSS). The purpose in developing this guide was to provide a document with comprehensive, detailed instructions that will reduce the time required by the user to learn the MAPSS program. The document was developed and in a sense beta tested with the help of several in-house colleagues. These colleagues provided invaluable comments after carefully comparing the instruction contained herein with the MAPSS program. In spite of the rigorous review, minor errors are likely since this is the first version. It is also possible that further description in some areas would aid users, as yet, unfamiliar with MAPSS. Furthermore, if history holds, MAPSS like DYNGEN [4] and DIGTEM [5] before it will likely become a "work horse" analytical tool for turbine engine controls research for the next 10-20 years. It is extremely likely that during that time it will be necessary to modify MAPSS to improve and extend its capabilities. The benefits and utility of those modifications should necessarily be identified and discussed in future updates to this document, or one like it. For these reasons, please feel free to contact the authors below with any comments or suggestions for improvements to this user guide, or to the MAPSS program itself.

Khary I. Parker: 216-433-8442; Khary.I.Parker@nasa.gov

Kevin J. Melcher: 216-433-3743; Kevin.J.Melcher@nasa.gov

References

1. Parker, K.I., and Guo, T.-H., "Development of a Turbofan Engine Simulation in a Graphical Simulation Environment," NASA/TM-2003-212543, August 2003.
2. MATLAB, "Using MATLAB," The MathWorks, Inc., 2000.
3. Simulink, "Using Simulink," The MathWorks, Inc., 2002.
4. Sellers, J.F., and Daniele, C.J., "DYNGEN: A Program for Calculating Steady-State and Transient Performance of Turbojet and Turbofan Engines," NASA-TN-D-7901, April 1975.
5. Daniele, C.J., Krosel, S.M., Szuch, J.R., and Westerkamp, E.J., "Digital Computer Program for Generating Dynamic Turbofan Engine Models (DIGTEM)," NASA TM-83446, September 1983.

Appendix A

Model Constants

Sensor:

xn2c_bias: Fan rotor speed bias
xn25c_bias: Core rotor speed bias
t27c_bias: Booster hub temperature bias
t27dc_bias: Booster tip temperature bias
t3c_bias: HPC exit temperature bias
t5bc_bias: LPT blade temperature bias
t56c_bias: LPT exit temperature bias
ps15c_bias: BP duct static pressure bias
p16c_bias: BP duct total pressure bias
ps21c_bias: Booster inlet static pressure bias
p27c_bias: Booster hub total pressure bias
p27dc_bias: Booster tip total pressure bias
ps3c_bias: HPC exit static pressure bias
ps56c_bias: Mixing plain static pressure bias
p58c_bias: Mixer exit total pressure bias

Controller:

iswoper: SM2 mode switch (1 | {0})
isiso: SISO mode switch
pcn2r_switch: Percent corrected rotor speed limit
stp27d_bias: Booster duct vane angle bias (deg.)
stp27_bias: Booster hub vane angle bias (deg.)
stp2_bias: Fan IGV angle bias (deg.)
sm2dem: Fan stall margin demand
taubw: PC lagged time constant
min_ps3: Min. burner static pressure (psi)

Actuator:

wf36_bias: Burner fuel flow actuator bias
stp27d_bias: Booster duct vane angle actuator bias (deg.)
stp27_bias: Booster hub vane angle actuator bias (deg.)
stp2_bias: Fan IGV angle actuator bias (deg.)
a8_bias: Nozzle throat area actuator bias
a16_bias: Rear VABI area actuator bias
a14_bias: Forward blocker door area actuator bias

Health Parameters:

zsw2: Fan flow scalar
sedm2: Fan flow efficiency scalar
zsw7d: Booster tip flow scalar
sedm7d: Booster tip efficiency scalar
zsw27: Booster hub flow scalar
sedm27: Booster hub efficiency scalar
zsw41: HPT flow scalar
zse41: HPT efficiency scalar
zsw49: LPT flow scalar
zse49: LPT efficiency scalar

Acceleration:

zpxwh: Core rotor power extraction factor
xjhp: Core rotor speed multiplier
xjlp: Fan rotor speed multiplier
c70959: Rotor acceleration constant

Appendix B

Model Output Variables

These variables are listed in the order in which they appear in mdlOut:

Actuator Positions:

1. *wf36n*: Main Burner Fuel Flow (lbm/hr)
2. *za8*: Variable Nozzle Area (in.²)
3. *za16*: Rear BP Door Variable Area (in.²)
4. *za14*: Forward BP Door Var. Area(in.²)
5. *zstp2*: Fan IVG Angle (degrees)
6. *zstp27*: HPC VSV Angle (degrees)
7. *zstp27d*: Booster IVG Angle (degrees)

Temperature and Pressure:

8. *tamb*: Ambient Temperature (°R)
9. *pamb*: Ambient Pressure (psia)
10. *t2*: Fan Inlet Temperature (°R)
11. *p2*: Fan Inlet Pressure (psia)

Component Power Usage:

12. *pw2*: Fan Power Consumption (W)
13. *pw27d*: Booster Power Consumption (W)
14. *pw27*: HPC Power Consumption (W)
15. *pw4*: HPT Power Output (W)
16. *pw48*: LPT Power Output (W)

LPT Error:

17. *yl49r*: LPT Loss Difference (Act. - Guess)
18. *yffl*: LPT Flow Function

HPT Error:

19. *yl41r*: HPT Loss Difference (Act. - Guess)
20. *yffh*: HPT Flow Function

Fan Exit:

21. *pnrm2*: % Corrected Fan Rotor Speed (%)
22. *p21aq*: Avg. Fan Press. Ratio
23. *p21a*: Avg. Fan Exit Press. (psia)
24. *t21a*: Avg. Fan Exit Temp. (°R)
25. *w2*: Physical Fan Flow (lbm/s)
26. *w2r*: Corr. Fan Flow (lbm/s)
27. *e2*: Fan Efficiency

HPC Exit:

28. *t3*: HPC Total Exit Temp. (°R)
29. *h3*: HPC Exit Enthalpy ((lbf-ft)/lbm)
30. *p3*: HPC Total Exit Pressure (psia)
31. *w27r*: HPC Corrected Flow (lbm/s)

32. *yw27*: HPC Actual and Map Flow Error (lbm/s)
33. *p27aq*: HPC Pressure Ratio
34. *e27*: HPC Efficiency
35. *pnrm27*: % Corrected HPC Rotor Speed (%)
36. *dstp27*: Difference in VSV Angle degrees)
37. *wrb27*: HPC Min. Loss Corrected Flow (lbm/s)
38. *pqb27*: HPC Min. Loss Pressure Ratio

Booster Exit:

39. *w27dr*: Booster Corrected Flow (lbm/s)
40. *w3d*: Booster Exit Flow (lbm/s)
41. *t3d*: Avg. Booster Exit Temp. (°R)
42. *p3d*: Avg. Booster Exit Press. (psia)
43. *p27daq*: Booster Ideal Pressure Ratio
44. *yw27d*: Booster Actual and Map Flow Error (lbm/s)
45. *e27d*: Booster Efficiency
46. *pnr27d*: % Corrected Booster Rotor Speed (%)
47. *wrb27d*: Booster Min. Loss Corr. Flow (lbm/s)
48. *pqb27d*: Booster Ideal Pressure Ratio

Bypass Stream:

49. *p14*: Booster BP Total Press. (psia)
50. *w14*: Booster BP Flow (lbm/s)
51. *t14*: Booster BP Total Temperature (°R)
52. *ps21a*: Fan Exit Static Press. (psia)

Duct Stream:

53. *h27d*: Duct Stream Enthalpy ((lbf-ft)/lbm)
54. *t27d*: Duct Stream Total Temperature (°R)
55. *p21d*: Duct Stream Fan Exit Total Pressure (psia)
56. *p27d*: Duct Stream Total Pressure (psia)
57. *w27d*: Duct Stream Mass Flow (lbm/s)

Core Stream:

58. *h27*: HPC Face Enthalpy ((lbf-ft)/lbm)
59. *t27*: HPC Face Total Temp. (°R)

- 60. *p27*: HPC Face Total Press. (psia)
- 61. *w27*: HPC Face Mass Flow (lbm/s)

Burner Exit:

- 62. *p4*: Burner Exit Total Pressure (psia)
- 63. *t4*: Burner Exit Total Temperature (°R)
- 64. *h4*: Burner Exit Enthalpy per pound of gas ((lb-ft)/lbm)
- 65. *dtm_{pc}*: Heat Soak Derivative
- 66. *w3*: HPC Exit Mass Flow (lbm/s)
- 67. *far4*: Burner Fuel/Air Ratio
- 68. *w4*: Burner Exit Total Mass Flow (lbm/s)
- 69. *wfc*: Burner Fuel Flow Coefficient (lbm/hr)

Turbine Exit:

- 70. *h5*: LPT Exit Enthalpy ((lb-ft)/lbm)
- 71. *t5*: LPT Exit Temp. (°R)
- 72. *far5*: LPT Fuel/Air Ratio
- 73. *w5*: LPT Exit Mass Flow (lbm/s)
- 74. *p5*: LPT Exit Press. (psia)

Mixer Exit:

- 75. *t58*: Mixer Exit Temp. (°R)
- 76. *w58*: Mixer Exit Mass Flow (lbm/s)
- 77. *p6*: AB Inlet Total Pressure (psia)
- 78. *ps6*: AB Inlet Static Pressure (psia)
- 79. *p58*: Mixer Exit Pressure (psia)
- 80. *ps56*: LPT Static Pressure at Mixer (psia)

Nozzle Exit:

- 81. *t7*: Exhaust Nozzle Throat Temp. (°R)
- 82. *w7*: Exhaust Nozzle Throat Mass Flow (lbm/s)
- 83. *far7*: Exhaust Nozzle Throat Fuel/Air Ratio
- 84. *p7qam*: Exhaust Nozzle Exit Pressure Ratio

Afterburner Exit:

- 85. *w68*: AB Mass Flow (lbm/s)
- 86. *wfctot*: AB Total Fuel Flow Coefficient (lbm/s)
- 87. *er68*:
- 88. *far68*: AB Fuel/Air Ratio
- 89. *h68*: AB Enthalpy ((lb-ft)/lbm)

BP Mixing Plane:

- 90. *w16*: Total AB Liner Cooling Mass Flow (lbm/s)
- 91. *ps156*: BP Mixing Plane static pressure (psia)
- 92. *p16*: Pressure at Variable AB Liner Area (psia)

BP Station 1.5 Exit:

- 93. *h15*: BP Enthalpy ((lb-ft)/lbm)
- 94. *t15*: BP Total Temperature (°R)
- 95. *w15*: BP Mass Flow (lbm/s)
- 96. *p15*: BP Total Pressure (psia)
- 97. *xmom15*: BP Momentum (lbm(ft/s))

BP Station 1.5.4 Exit:

- 98. *h154*: BP Exit Enthalpy ((lb-ft)/lbm)
- 99. *t154*: BP Exit Total Temperature (°R)
- 100. *w154*: BP Exit Mass Flow (lbm/s)
- 101. *p154*: BP Exit Total Pressure (psia)

Cooling Airflow:

- 102. *wcl41*: HPT Inlet Bleed Flow (lbm/s)
- 103. *wcl42*: HPT Exit Bleed Flow (lbm/s)
- 104. *wcl5*: LPT Exit Bleed Flow (lbm/s)
- 105. *wcl69*: AB Bleed Flow (lbm/s)
- 106. *wcl54*: HPC Bleed Flow (lbm/s)

Cooling Enthalpy:

- 107. *hcl54*: HPC Bleed Enthalpy ((lb-ft)/lbm)
- 108. *hcl69*: AB Bleed Enthalpy ((lb-ft)/lbm)

Duct Inlet Pressure:

- 109. *p3f*:

Compressor Stall:

- 110. *sm27d*: Booster Stall Margin (%)
- 111. *sm27*: HPC Stall Margin (%)

Fan Stall:

- 112. *sm2b*: Fan Stall Margin (%)
- 113. *spsd2*: Fan Stall Line Scalar due to Distortion (%)

Update Parameters:

- 114. *bp25dg*: HPC BP Duct Flow Ratio
- 115. *bpr14g*: Forward Duct BP Ratio

- 116. *gh2g*: Fan Off Backbone Work
($\text{g}_c \cdot \text{lbm}(\text{ft}^2/\text{s}^2)$)
- 117. *gh27dg*: Booster Off Backbone Work
($\text{g}_c \cdot \text{lbm}(\text{ft}^2/\text{s}^2)$)
- 118. *gh27g*: HPC Off Backbone Work
($\text{g}_c \cdot \text{lbm}(\text{ft}^2/\text{s}^2)$)
- 119. *psdqg*: Duct Static Pressure Guess (psia)
- 120. *dpfhpq*: Turbine Pressure Split (psia)
- 121. *hl49rg*: LPT Loss
- 122. *hl41rg*: HPT Loss
- 123. *p5q2g*: LPT Exit Pressure Guess (psia)

HPC Damage Bleeds:

- 124. *h3bdm*: HPC Damage Bleed Enthalpy
($(\text{lbf} \cdot \text{ft})/\text{lbm}$)
- 125. *t3bdm*: HPC Damage Bleed Temperature
($^{\circ}\text{R}$)
- 126. *w3bdm*: HPC Damage Bleed Mass Flow
(lbm/s)
- 127. *p3bdm*: HPC Damage Bleed Pressure
(psia)

HPT Exit:

- 128. *h42p*: HPT Exit Enthalpy ($(\text{lbf} \cdot \text{ft})/\text{lbm}$)
- 129. *e42*: HPT Efficiency
- 130. *wrb41*: HPT Min. Loss Flow
- 131. *dgn41*: HPT Backbone Loading (lbf)

LPT Exit:

- 132. *h5p*: LPT Exit Enthalpy ($(\text{lbf} \cdot \text{ft})/\text{lbm}$)
- 133. *e49*: LPT Efficiency

Sensors:

- 134. *fn*: Net Thrust (lbf)
- 135. *sfc*: Uninstalled Thrust Specific Fuel
Consumption ($\text{lbm}/(\text{hr} \cdot \text{lbf})$)
- 136. *sm2*: Fan Stall Margin (%)
- 137. *sm27d*: Booster Stall Margin (%)
- 138. *sm27*: HPC Stall Margin (%)
- 139. *xn2c*: LP Spool Sensor (RPM)
- 140. *xn25c*: HP Spool Sensor (RPM)
- 141. *pcn2*: % LP Spool Rotor Speed (%)
- 142. *ps21c*: Fan Exit Static Press. Sensor (psia)
- 143. *p27dc*: Booster Inlet Press. Sensor (psia)
- 144. *t27dc*: Booster Inlet Temp. Sensor ($^{\circ}\text{R}$)
- 145. *p27c*: HPC Inlet Press. Sensor (psia)
- 146. *t27c*: HPC Inlet Temp. Sensor ($^{\circ}\text{R}$)
- 147. *t3c*: HPC Exit Temp Sensor ($^{\circ}\text{R}$)
- 148. *ps15c*: BP Duct Static Press. Sensor at
Mixer (psia)
- 149. *p16c*: Mixer Inlet Press. Sensor (psia)
- 150. *ps3c*: HPC Exit Static Press. Sensor (psia)
- 151. *t5bc*: LPT Blade Temp. Sensor ($^{\circ}\text{R}$)
- 152. *t56c*: LPT Exit Temp. Sensor ($^{\circ}\text{R}$)
- 153. *ps56c*: LPT Exit Static Press. at Mixer
(psia)
- 154. *p58c*: Mixer Exit Press. Sensor (psia)
- 155. *p16qp56*: Mixer Press. Ratio Sensor

States:

- 156. *xnl*: LP Spool Speed (RPM)
- 157. *xnh*: HP Spool Speed (RPM)
- 158. *tmpc*: Heat Soak Temp. ($^{\circ}\text{R}$)

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 2004		3. REPORT TYPE AND DATES COVERED Technical Memorandum
4. TITLE AND SUBTITLE The Modular Aero-Propulsion System Simulation (MAPSS) Users' Guide			5. FUNDING NUMBERS WBS-22-704-30-01	
6. AUTHOR(S) Khary I. Parker and Kevin J. Melcher				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration John H. Glenn Research Center at Lewis Field Cleveland, Ohio 44135-3191			8. PERFORMING ORGANIZATION REPORT NUMBER E-14441	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-2004-212968	
11. SUPPLEMENTARY NOTES Responsible person, Khary I. Parker, organization code 5530, 216-433-8442.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Categories: 07 and 08 Available electronically at http://gltrs.grc.nasa.gov This publication is available from the NASA Center for AeroSpace Information, 301-621-0390.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The Modular Aero-Propulsion System Simulation is a flexible turbofan engine simulation environment that provides the user a platform to develop advanced control algorithms. It is capable of testing the performance of control designs on a validated and verified generic engine model. In addition, it is able to generate state-space linear models of the engine model to aid in controller design. The engine model used in MAPSS is a generic high-pressure ratio, dual-spool, low-bypass, military-type, variable cycle turbofan engine with a digital controller. MAPSS is controlled by a graphical user interface (GUI) and this guide explains how to use it to take advantage of the capabilities of MAPSS.				
14. SUBJECT TERMS Digital simulation; Dynamic system; Models; Computer aided design; Control system design; Engine control			15. NUMBER OF PAGES 29	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	